

# Data warehouse in Clothing Shop

## 1. Business Description:

At Girls Warehouse, we take pride in being at the forefront of innovation and efficiency in the realm of product storage and distribution. Our warehouse is more than just a physical space; it is the heartbeat of our operations, meticulously designed to ensure a seamless flow of goods and information.

### **Strategic Location:**

Situated in [City/Region], our warehouse boasts a strategic location that facilitates swift and cost-effective transportation. This strategic positioning enables us to reach our customers across regions promptly, meeting their needs with unparalleled efficiency.

### **Cutting-edge Technology:**

Equipped with the latest in warehouse technology, we leverage advanced inventory management systems, automated order processing, and real-time tracking. This not only enhances our operational efficiency but also provides our clients with transparency and visibility into their supply chain.

### **Diverse Product Portfolio:**

Girls Warehouse is home to a diverse range of products, each meticulously categorized and stored for easy retrieval and distribution. From the latest fashion trends to everyday essentials, our warehouse accommodates the dynamic nature of our product catalog.

### **Customer-Centric Approach:**

Our commitment to customer satisfaction is embedded in every aspect of our warehouse operations. The GIRLS\_CUSTOMER table in our database captures essential customer information, allowing us to tailor our services and maintain strong relationships with our clients.

### **Sustainable Practices:**

Girls Warehouse is not just a space for storage; it is a testament to our commitment to sustainability. Our eco-friendly initiatives include energy-efficient lighting, recycling programs, and environmentally conscious packaging, ensuring that our operations are not only efficient but also environmentally responsible.

### **Secure and Controlled Environment:**

Security is paramount at Girls Warehouse. With state-of-the-art surveillance systems, access control measures, and a dedicated security team, we guarantee the safety and integrity of the products entrusted to us.

### **Collaborative Partnerships:**

We recognize that the success of Girls Warehouse is intertwined with the success of our partners. By fostering collaborative relationships with suppliers, manufacturers, and logistics partners, we create a network that ensures the seamless flow of goods from production to delivery.

## 2. Use Cases:

### **Real-time Inventory Management:**

Utilizing our advanced inventory management system, the client can access real-time data, ensuring accurate stock levels, minimizing stockouts, and optimizing replenishment strategies.

### **Efficient Order Fulfillment:**

The Girls Warehouse's automated order processing system ensures swift order picking, packing, and shipping, meeting increased demand without compromising on accuracy or delivery timelines.

### **Seasonal Inventory Planning:**

Leveraging historical data from the GIRLS\_TIME\_DM table, our analytics tools help the client analyze trends, optimize stocking levels, and plan for seasonal fluctuations, ensuring efficient inventory turnover.

### **Sustainable Supply Chain Practices:**

Girls Warehouse incorporates eco-friendly measures, providing the client with an environmentally conscious distribution solution. This not only meets corporate social responsibility goals but also attracts eco-conscious consumers.

## 3. Dimensions:

Customer Dimension (GIRLS\_CUSTOMER):

Attributes:

customer\_ID: Unique identifier for each customer.

customer\_name: First name of the customer.

customer\_surname: Last name of the customer.

customer\_address: Address of the customer.

customer\_contact\_number: Contact number of the customer.

customer\_email: Email address of the customer.

country\_ID: Unique identifier for the customer's country.

country\_name: Name of the customer's country.

Anastasiia Skrypnyk

Product Dimension (GIRLS\_PRODUCTS):

Attributes:

product\_ID: Unique identifier for each product.

product\_name: Name of the product.

product\_description: Description of the product.

category\_ID: Unique identifier for the product category.

category\_name: Name of the product category.

category\_description: Description of the product category.

Time Dimension (GIRLS\_TIME\_DM):

Attributes:

year\_ID: Unique identifier for the year.

year\_name: Name of the year.

year\_end\_date: End date of the year.

month\_ID: Unique identifier for the month.

month\_name: Name of the month.

day\_ID: Unique identifier for the day.

day\_name: Name of the day.

Fact Table:

Orders table is the fact table which contain the foreign key and the measurements columns like quantity and amount to calculate any information across all tables if needed.

Orders Fact Table (GIRLS\_ORDERS):

Attributes:

customer\_ID: Foreign key linking to GIRLS\_CUSTOMER, representing the customer placing the order.

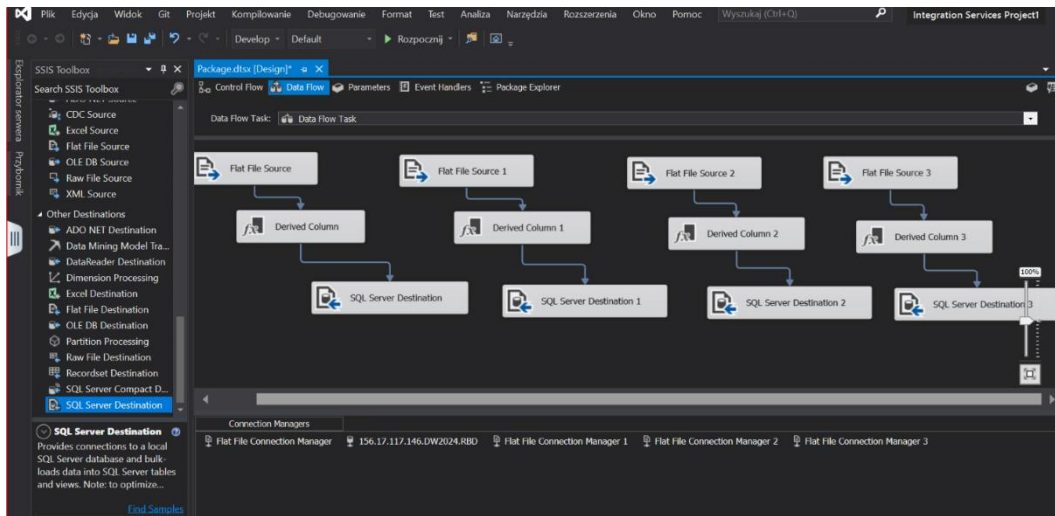
product\_ID: Foreign key linking to GIRLS\_PRODUCTS, representing the product being ordered.

day\_ID: Foreign key linking to GIRLS\_TIME\_DM, representing the day the order was placed.

ordered\_quantity: The quantity of the product ordered.

order\_total: The total cost of the order.

## 4. ETL process created in Integration Services Project in Visual Studio



## a. Products:

**Flat File Source Editor**

Configure the properties used to connect to and obtain data from a text file.

Connection Manager: Flat file connection manager: Flat File Connection Manager

☐ Retain null values from the source as null values in the data flow

**Data View**

Sample data (up to first 200 rows)

product_ID	product_name	product_desc...	category_ID	category_name
'001'	'Basic T-shirt'	'Cotton T-shirt'	'2023_4'	'T-shirt'
'002'	'Long Sleeve...	'Cotton T-shirt'	'2023_4'	'T-shirt'
'003'	'Neck Bodysuit'	'Polyester Bo...	'Bodysuit'	'Woman Bod...
'004'	'Short Sleeve...	'Polyester Cr...	'2023_2'	'Top'
'005'	'Satin Halter ...	'Polyester Top'	'2023_2'	'Top'

**Derived Column Transformation Editor**

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

**Variables and Parameters**

- Columns
  - product\_ID
  - product\_name
  - product\_description
  - category\_ID
  - category\_name
  - category\_description

**Mathematical Functions**

- String Functions
- Date/Time Functions
- NULL Functions
- Type Casts
- Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type
product_ID	Replace 'product_ID'	[product_ID]	string [DT_STR]
product_name	Replace 'product_name'	[product_name]	string [DT_STR]
product_description	Replace 'product_desc...	[product_description]	string [DT_STR]
category_ID	Replace 'category_ID'	[category_ID]	string [DT_STR]
category_name	Replace 'category_na...	[category_name]	string [DT_STR]
category_description	Replace 'category_des...	[category_description]	string [DT_STR]

**SQL Destination Editor**

Configure the properties used to bulk copy data into a local instance of the Database Engine.

Connection Manager: Mappings Advanced

Specify a connection manager, data source, or data source view, and select the table or the view into which the data is copied. Click New to create a new table or view.

Connection manager: 156.17.117.146.DW2024.RBD

Use a table or view: [dbo].[GIRLS\_PRODUCTS]

Preview

b. Customer:

Flat File Source Editor

Configure the properties used to connect to and obtain data from a text file.

Connection Manager  
Columns  
Error Output

Flat file connection manager:  
Flat File Connection Manager 3  

New...

☐ Retain null values from the source as null values in the data flow

Preview...

Data View

Sample data (up to first 200 rows)

customer_ID	customer_na...	customer_su...	customer_a...	customer_co...
'h1weu1234'	'Alice	'Boston'	'Kamienina 1...	'50-545 Wro...
'bank123'	'Margaret'	'Flower'	'Stone 2	'69-908 Lond...
'hjul345'	'Agata'	'Byk'	'Zlota 2	'43-098 Wars...
'jhgl123'	'Marcin'	'Cukier'	'Modrzewiow...	'32-908 Krak...
'jkol390'	'Jadwiga'	'Kol'	'Partyzantow 3	'59-900 Gy...
'heio8978'	'Olaf'	'Snow'	'Lodowa 76/12	'89-900 Gda...

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Variables and Parameters  
Columns  
customer\_ID  
customer\_name  
customer\_surname  
customer\_address  
customer\_contact\_number  
customer\_email  
country\_ID  
country\_name

Mathematical Functions  
String Functions  
Date/Time Functions  
NULL Functions  
Type Casts  
Operators

Description:

Derived Column Name	Derived Column	Expression	Data Type	Le
customer_ID	Replace 'customer_ID'	[customer_ID]	string [DT_STR]	50
customer_name	Replace 'customer_na...	[customer_name]	string [DT_STR]	50
customer_surname	Replace 'customer_sur...	[customer_surname]	string [DT_STR]	50
customer_address	Replace 'customer_ad...	[customer_address]	string [DT_STR]	50
customer_contact_nu...	Replace 'customer_co...	[customer_contact_number]	string [DT_STR]	50
customer_email	Replace 'customer_em...	[customer_email]	string [DT_STR]	50
country_ID	Replace 'country_ID'	[country_ID]	string [DT_STR]	50
country_name	Replace 'country_name'	[country_name]	string [DT_STR]	50

SQL Destination Editor

Configure the properties used to bulk copy data into a local instance of the Database Engine.

Connection Manager  
Mappings  
Advanced

Specify a connection manager, data source, or data source view, and select the table or the view into which the data is copied. Click New to create a new table or view.

Connection manager:  
156.17.117.146.DW2024.RBD  

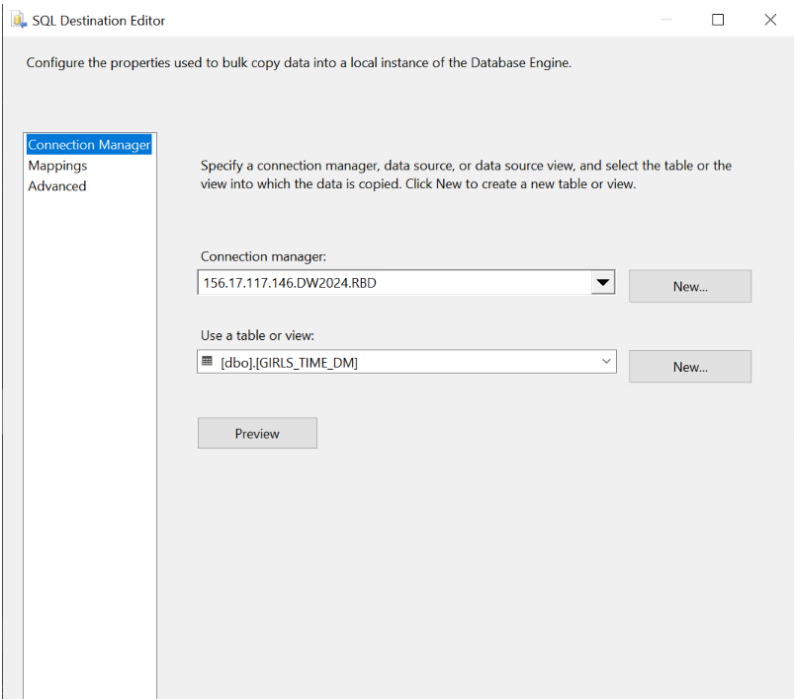
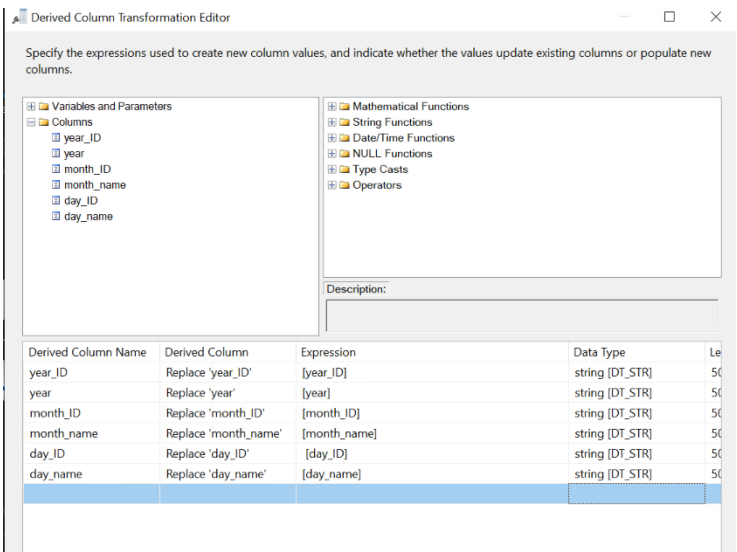
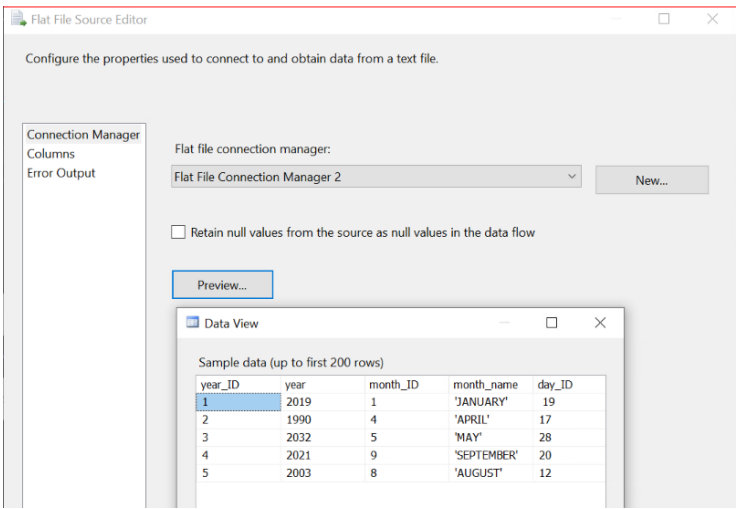
New...

Use a table or view:  
[dbo].[GIRLS\_CUSTOMER]  

New...

Preview

c. Time\_dm:



d. Orders:

Flat File Source Editor

Configure the properties used to connect to and obtain data from a text file.

Connection Manager  
Columns  
Error Output

Flat file connection manager:  
Flat File Connection Manager 1  

Retain null values from the source as null values in the data flow

Preview...

Data View

Sample data (up to first 200 rows)

customer_ID	product_ID	day_ID	ordered_qua...	order_total
'nieveu1234'	'001'	19	78	1500
'bank123'	'003'	28	87	1890
'nju1345'	'005'	20	87	6542
'jhgc123'	'004'	12	23	89624
'nelo8978'	'002'	17	1	12345

Derived Column Transformation Editor

Specify the expressions used to create new column values, and indicate whether the values update existing columns or populate new columns.

Variables and Parameters  
Columns  
customer\_ID  
product\_ID  
day\_ID  
ordered\_quantity  
order\_total

(DT\_BOOL)  
(DT\_NUMERIC, «precision», «scale»)  
(DT\_DECIMAL, «scale»)  
(DT\_CY)  
(DT\_GUID)  
(DT\_BYTES, «length»)  
(DT\_DBDATE)  
(DT\_DBTIME)  
(DT\_DBTIME2, «scale»)  
(DT\_DBTIMESTAMP)  
(DT\_DBTIMESTAMP2, «scale»)

Description:

Derived Column Name	Derived Column	Expression	Data Type	Le
customer_ID	Replace 'customer_ID'	[customer_ID]	string [DT_STR]	50
product_ID	Replace 'product_ID'	[product_ID]	string [DT_STR]	50
day_ID	Replace 'day_ID'	[day_ID]	string [DT_STR]	50
ordered_quantity	Replace 'ordered_qua...	(DT_NUMERIC, «precision», «scale») [ordered_qua...	string [DT_STR]	50
order_total	Replace 'order_total'	(DT_NUMERIC, «precision», «scale») [order_total]	string [DT_STR]	50

SQL Destination Editor

Configure the properties used to bulk copy data into a local instance of the Database Engine.

Connection Manager  
Mappings  
Advanced

Specify a connection manager, data source, or data source view, and select the table or the view into which the data is copied. Click New to create a new table or view.

Connection manager:  
156.17.117.146.DW2024.RBD  
New...

Use a table or view:  
{dbo}.[GIRLS\_ORDERS]  
New...

Preview

## 5. SQL code used to create objects

```
CREATE TABLE CUSTOMER (  
    customer_ID VARCHAR(10),  
    customer_name VARCHAR(15),  
    customer_surname VARCHAR(30),  
    customer_address VARCHAR(150),  
    customer_contact_number VARCHAR(10),  
    customer_email VARCHAR(100)  
    country_ID VARCHAR(10),  
    country_name VARCHAR(50)  
);
```

```
CREATE TABLE PRODUCT(  
    product_ID VARCHAR(10),  
    product_name VARCHAR(50),  
    product_description VARCHAR(150),  
    category_ID VARCHAR(10),  
    category_name VARCHAR(50)  
    category_description VARCHAR(100)  
);
```

```
CREATE TABLE TIME_DM(  
    year_ID VARCHAR(10),  
    year_name VARCHAR(50),  
    year_end_date VARCHAR(10),  
    month_ID VARCHAR(10),  
    month_name VARCHAR(20),  
    day_ID VARCHAR(10),  
    day_name VARCHAR(20)  
);
```

```
CREATE TABLE ORDER(  
    customer_ID VARCHAR(10),  
    product_ID VARCHAR(10),  
    day_ID VARCHAR(10),  
    ordered_quantity INTEGER(10),  
    order_total INTEGER(10)  
);
```

## 6. INSERT statements in SQL

```
CREATE TABLE GIRLS_TIME_DM
```



Anastasiia Skrypnyk

```
(  
    year_ID NUMBER,  
    year  NUMBER,  
    month_ID NUMBER,  
    month_name VARCHAR(30),  
    day_ID NUMBER,  
    day_name VARCHAR(30)  
);
```

```
INSERT INTO GIRLS_TIME_DM (year_ID, year, month_ID, month_name, day_ID, day_name) VALUES  
( '1', '2019', '1', 'JANUARY', '19', 'MONDAY');
```

```
INSERT INTO GIRLS_TIME_DM (year_ID, year, month_ID, month_name, day_ID, day_name) VALUES  
( '2', '1990', '4', 'APRIL', '17', 'WEDNESDAY');
```

```
INSERT INTO GIRLS_TIME_DM (year_ID, year, month_ID, month_name, day_ID, day_name) VALUES  
( '3', '2032', '5', 'MAY', '28', 'FRIDAY');
```

```
INSERT INTO GIRLS_TIME_DM (year_ID, year, month_ID, month_name, day_ID, day_name) VALUES  
( '4', '2021', '9', 'SEPTEMBER', '20', 'MONDAY');
```

```
INSERT INTO GIRLS_TIME_DM (year_ID, year, month_ID, month_name, day_ID, day_name) VALUES  
( '5', '2003', '8', 'AUGUST', '12', 'THURSDAY');
```

CREATE TABLE **GIRLS\_ORDER**

```
(  
    customer_ID VARCHAR(30),  
    product_ID  VARCHAR(30),  
    day_ID      NUMBER,  
    ordered_quantity NUMBER,  
    order_total  NUMBER  
);
```

```
INSERT INTO GIRLS_ORDER (customer_ID, product_ID, day_ID, ordered_quantity, order_total) VALUES  
( 'nieweu1234', '001', 19, 78, 1500);
```

```
INSERT INTO GIRLS_ORDER (customer_ID, product_ID, day_ID, ordered_quantity, order_total) VALUES  
( 'bank123', '003', 28, 87, 1890);
```

Anastasiia Skrypnyk

```
INSERT INTO GIRLS_ORDER (customer_ID, product_ID, day_ID, ordered_quantity, order_total) VALUES ('njui345', '005', 20, 87, 6542);
```

```
INSERT INTO GIRLS_ORDER (customer_ID, product_ID, day_ID, ordered_quantity, order_total) VALUES ('jhgk123', '004', 12, 23, 89624);
```

```
INSERT INTO GIRLS_ORDER (customer_ID, product_ID, day_ID, ordered_quantity, order_total) VALUES ('neio8978', '002', 17, 1, 12345);
```

CREATE TABLE **GIRLS\_PRODUCT**

```
(  
    product_ID  VARCHAR(30),  
    product_name    VARCHAR(30),  
    product_description VARCHAR(30),  
    category_ID  VARCHAR(30),  
    category_name    VARCHAR(30),  
    category_description VARCHAR(150)  
);
```

```
INSERT INTO GIRLS_PRODUCT (product_ID, product_name, product_description, category_ID,  
category_name, category_description) VALUES ('001', 'Basic T-shirt', 'Cotton T-shirt', '2023_4', 'T-shirt', 'Woman T-shirt');
```

```
INSERT INTO GIRLS_PRODUCT (product_ID, product_name, product_description, category_ID,  
category_name, category_description) VALUES ('002', 'Long Sleeve T-shirt', 'Cotton T-shirt', '2023_4', 'T-shirt', 'Woman T-shirt');
```

```
INSERT INTO GIRLS_PRODUCT (product_ID, product_name, product_description, category_ID,  
category_name, category_description) VALUES ('003', 'Neck Bodysuit', 'Polyester Bodysuit', '2023_2', 'Bodysuit', 'Woman Bodysuits');
```

```
INSERT INTO GIRLS_PRODUCT (product_ID, product_name, product_description, category_ID,  
category_name, category_description) VALUES ('004', 'Short Sleeve Crop Top', 'Polyester Crop Top', '2023_2', 'Top', 'Woman Crop Top');
```

```
INSERT INTO GIRLS_PRODUCT (product_ID, product_name, product_description, category_ID,  
category_name, category_description) VALUES ('005', 'Satin Halter Top', 'Polyester Top', '2023_2', 'Top', 'Woman Halter Top');
```

Anastasiia Skrypnyk

CREATE TABLE **GIRLS\_CUSTOMER**

```
(  
    customer_ID VARCHAR(30),  
    customer_name    VARCHAR(50),  
    customer_surname VARCHAR(50),  
    customer_address VARCHAR(150),  
    customer_contact_number VARCHAR(12),  
    customer_email    VARCHAR(50),  
    country_ID  VARCHAR(10),  
    country_name    VARCHAR(30)  
);
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'niweu1234', 'Alice', 'Boston', 'Kamienna 190/23 50-545 Wroclaw', '678876567',  
'aliceboston@gmail.com', 1, 'Poland');
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'bank123', 'Margaret', 'Flower', 'Stone 2 69-908 London', '909876543', 'margaret@gmail.com', 'CA'  
'Canada');
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'njui345', 'Agata', 'Byk', 'Zlota 2 43-098 Warszawa', '8765443123', 'agatka@gmail.com', 'GE',  
Germany');
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'jhkg123', 'Marcin', 'Cukier', 'Modrzewiowa 1/9 32-908 Krakow', '787787989',  
'marcincukier@gmail.com', 'CR', 'Czech Republic');
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'jkoi390', 'Jadwiga', 'Kot', 'Partyzantow 3 59-900 Gy', '786543425', 'jadzia@gmail.com', 'HL',  
'Holand');
```

```
INSERT INTO GIRLS_CUSTOMER (customer_ID, customer_name, customer_surname,  
customer_address, customer_contact_number, customer_email, country_ID, country_name) VALUES  
( 'neio8978', 'Olaf', 'Snow', 'Lodowa 76/12 89-900 Gdansk', '766552414', 'null', 'ICE', 'Iceland');
```